



TITLE:

An application of persistent homology on musical data (Geometry, Algebra and Combinatorics in Transformation group theory)

AUTHOR(S):

濱岡, 智恵; 川島, 正行

CITATION:

濱岡, 智恵 ...[et al]. An application of persistent homology on musical data (Geometry, Algebra and Combinatorics in Transformation group theory). 数理解析研究所講究録 2018, 2098: 8-14

ISSUE DATE:

2018-12

URL:

<http://hdl.handle.net/2433/251755>

RIGHT:

An application of persistent homology on musical data

岡山理科大学・総合情報研究科・情報科学専攻 濱岡 智恵

岡山理科大学・総合情報学部・情報科学科 川島 正行

Tomoe Hamaoka, Masayuki Kawashima

Department of Information Science

Okayama University of Science

1 Introduction

近年では機械学習，深層学習の発展に伴ない Orpheus [1] や Google Magenta [2] などの自動作曲システムが開発されている．また作曲や編曲において，楽曲がもつ感情的な部分，即ち，悲しい曲，楽しい曲などについても多く議論されている [3]．

しかしこのような作曲システムにおいて、「楽曲」の定義が明確にされていないまま議論がされているように思える．本研究では，音の集合と，楽曲とは何が違うのかという部分を考察すること，即ち，音の進行の変化によっておこる現象について考察することを最終目標とする．

そのために本研究では楽曲の生成を，音源から行うのではなく，楽曲の音の数に注目し，その数をもとに多面体を構成し，その多面体を学習することで楽曲の特徴を解析する．

多面体を構成する可視化の手法としては3和音を平面上の三角形に見立て，それらを張り合わせた「Tonnetz」と呼ばれる平面図形を用いる．楽曲の MIDI ファイルを読み込み，そのクロマの数を元に Tonnetz から3次元内の多面体を構成することによって行う．またバーコードを計算することによってその多面体のトポロジーを調べる．本研究では，この手順を Windows10 において Python 2.7 (Python ソフトウェア財団) と OpenGL で実装を行なった．

2 Musical setting

2.1 音

本研究では楽曲を構成する最小単位「音」を，高さ $p \in \mathbb{R}_{\geq 0}$ と長さ $d \in \mathbb{R}_{\geq 0}$ によって作られる実数のペア $\mathbf{n} = (p, d) \in \mathbb{R}_{\geq 0}^2$ として定める．ここで音 $\mathbf{n} = (p, d)$ の高さ p は 1 秒間に何回振動するかを表し，単位は Hz を用いることが多い．また長さ d はその音がどのくらいの時間振動しているかを表し，ここでは CD の収録限界と考えられている 74 分 = 4440 秒以下の値を取るとする．音を，強弱を含めた 3 つのデータで定義する方法も検討されているが，ここでは自由度を広げるために，高さ と長さ のペアで考えるものとする．

2 つの音 $\mathbf{n}_1, \mathbf{n}_2$ に対してその周波数を用いて順序関係を定義する．即ち， $p_1 \geq p_2$ のとき $\mathbf{n}_1 \geq \mathbf{n}_2$ と定める．一般に \mathbf{n}_1 は \mathbf{n}_2 よりも高く聞こえる．また音 $\mathbf{n} = (p, d)$ に対して $\mathbf{n}' = (2p, d)$ を \mathbf{n} の 1 オクターブ上の音という．

どの音を基準にするかという議論も古くからされているが，現在では 1834 年に Stuttgart 会議で定められた特定の周波数を「標準高度」といい，440Hz を A の音をとする規格が基準として広く使われている．音の長さの基準は 1 であるとし，音名はこの A を基準として以下の 12 音で構成されている：

$$A = (440, 1)$$

$$A, \quad A^\sharp, \quad B, \quad C, \quad C^\sharp, \quad D, \quad D^\sharp, \quad E, \quad F, \quad F^\sharp, \quad G, \quad G^\sharp.$$

それぞれの音の高さは，公比が $\sqrt[12]{2}$ の等比数列をなすように関係つけられている．すなわち C の周波数を p とすると C^\sharp の周波数は $\sqrt[12]{2}p$ となっている．

一般に使用される 88 鍵ピアノは 27.5Hz ~ 4096Hz まで表現できる．もっとも低い 27.5Hz の音をオクターブ 0 のラといわれ A_0 で表すことも多い．これを基準とすると標準高度 440Hz はオクターブ 4 のラとなり A_4 と表すこともある．本研究では 88 鍵ピアノに従い，音 \mathbf{n} の高さ p は $27.5 \leq p \leq 4096$ を満たすとする．

2.2 和音

複数の音を同時にならすとき，それぞれ高さの関係からよく響くときがある．よく響くというのは曖昧ではあるが，ここではその音の組のことを「和音」または「コード」という．一般的によく使用されるのは 3 つの音の組 (n_1, n_2, n_3) から構成されている「三和音」と呼ばれるものである．その中でも 「長三和音（メジャー）」，「短三和音（マイナー）」と

呼ばれるものが主となって使われている．長三和音とは基準となる音に長3度と完全5度の音の組で構成される和音のことで基準の音が C の場合 $C = (C, E, G)$ と表される．短三和音とは基準となる音に短3度と完全5度の音の組で構成される和音のことで基準の音が C の場合 $Cm = (C, E^b, G)$ と表される．このメジャーコードとマイナーコードの音の関係を図示すると以下ようになる：

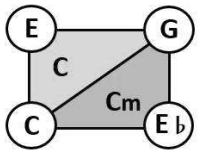


図 1 C メジャーコードと C マイナーコード

3 Tonnetz

Tonnetz とは音の高さの関係を平面上に表した格子状の図形のことである．1739 年に Leonhard Euler によって Tonnetz の原型が作られ，20 世紀に David Lewin らが発展させ，現在の形となった．一般には無限に広がるが，本研究では以下の有限範囲を用いる．Tonnetz は，それぞれの頂点から真上にいくと長3度，右に行くと短3度，右上に行くと完全5度を表現している．また，三角形の各頂点がコードを構成する音を示し，それぞれの三角形の中心に書いてあるものがコードの種類を示す．黄色の三角形がメジャーコード，青い三角形がマイナーコードである．

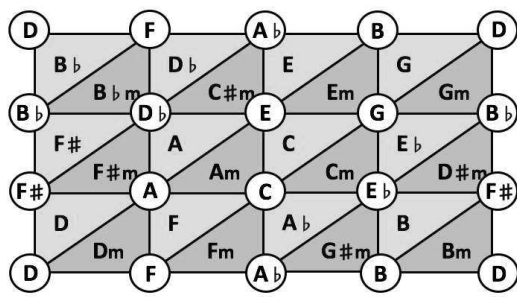


図 2 Tonnetz

この有限範囲の Tonnetz を有限 **Tonnetz** あるいは上下の辺と左右の辺を張り合わせる

ことによってトーラス構造を持つことから **Tours Tonnetz** といわれている。

有限 Tonnetz は三和音に対しての関係を表したもののだが、一般の五和音、七和音などの一般の和音に関しての表現の研究は Neo-Riemannian theory として現在も多くの研究がなされている。

4 多面体

ここでは \mathbb{R}^3 の中の多面体を定め、Tonnetz から多面体を構成する。多面体とは以下に定義される単体複体を \mathbb{R}^3 内に実現したものである。本研究で使用する記号は [4] に従った。

4.1 単体複体

\mathbb{R}^3 内の 0-単体とは「点」を表し、1-単体とは線分を表す。さらに 2-単体は三角形、3-単体は四面体を表す。 d -単体の境界は $(d-1)$ -単体になっていることに注意する。

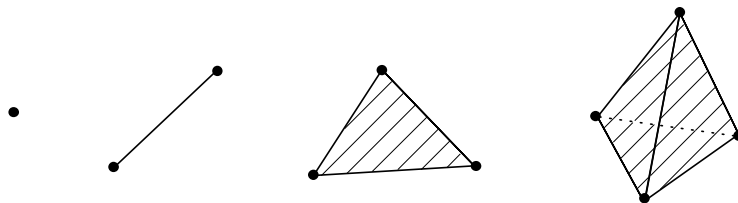


図 3 0-単体 1-単体 2-単体 3-単体

d -単体 σ の頂点を a_0, \dots, a_d とするとき σ を $\sigma = |a_0 \cdots a_d|$ と表すとき、その部分の頂点からなる単体 $\tau = |a_{i_0} \cdots a_{i_s}|$ を s -面といい、記号で $\tau \leq \sigma$ で表す。

単体複体とは、単体の集合 \mathcal{K} が以下の 2 つの条件を満たすものをいう。直感的には複数の単体が無理なく貼り合わされているということである。

1. 任意の単体 $\sigma \in \mathcal{K}$ に対して σ の面はまた \mathcal{K} に属する。
2. $\sigma, \tau \in \mathcal{K}$ で $\sigma \cap \tau \neq \emptyset$ のとき $\sigma \cap \tau$ は σ の面であり、 τ の面である。

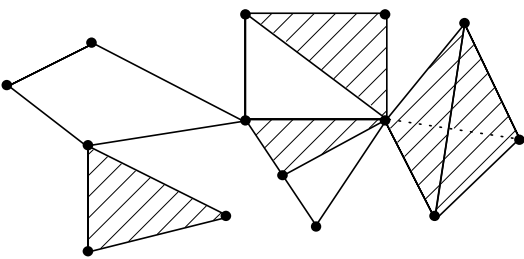


図 4 単体複体

\mathcal{K} を単体複体とすると、それに含まれるすべての単体の和集合を \mathcal{K} を多面体といい $|\mathcal{K}|$ で表す: $|\mathcal{K}| = \bigcup_{\sigma \in \mathcal{K}} \sigma$

4.2 Tonnetz から多面体の構成

与えられた楽曲の音の個数を高さとし、Tonnetz から \mathbb{R}^3 内の多面体を構成する。本研究では、Fig. 2 上にすべての音を表すため、オクターブは考慮せず同じ音として扱い、楽曲の分析は MIDI ファイルから以下の手順で音を解析する。

- 1. MIDI ファイルから MIDI メッセージを取得する。
- 2. MIDI メッセージからノート番号を抽出する。
- 3. そのノート番号をもとにクロマを計算する。
- 4. クロマからそれぞれの音の個数を計算しそれをもとに高さを決定する。
- 5. 4. の音の高さを元に、Fig. 2 から \mathbb{R}^3 の多面体を構成する。

この手順出来上がった多面体を変形 **Tonnetz** と呼ぶ。

例としてベートーベン作曲：ピアノソナタ第 14 番嬰ハ短調「月光」の冒頭部分の可視化を行う。以下は、月光の冒頭部分の楽譜である。



図 5 月光の冒頭部分

この MIDI ファイルを上位の手順で読み込むとクロマの個数は

$$\begin{aligned} C : 0, \quad C^b : 12, \quad D : 2, \quad E^b : 0, \quad E : 10, \quad F : 0 \\ F^\sharp : 4, \quad G : 0, \quad A^b : 8, A : 6, \quad B^b : 0, \quad B : 2 \end{aligned}$$

となる． この結果を変形 Tonnetz で表現すると以下のようなになる．

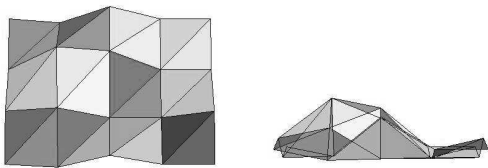


図 6 上と横から見た月光の変形 Tonnetz

5 多面体による可視化とバーコード

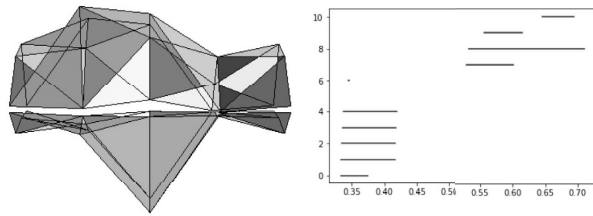
上述の手順に従って、いくつかの楽曲の可視化を行う．また出来上がった変形 Tonnetz からバーコードを計算しそのトポロジーを調べる．MIDI ファイルは web サイト [5] のファイルを用いた．作曲者，曲数は以下の通りである．

作曲者	曲数	作曲者	曲数	作曲者	曲数
js・バッハ	4	チャイコフスキー	1	シューマン	10
モーツァルト	6	ムソルグスキー	1	ドビュッシー	3
ショパン	5	スクリャービン	8	リスト	2
ブラームス	2	ラフマニノフ	1	スカルラティ	10
ベートーベン	18			計 13 人	71 曲

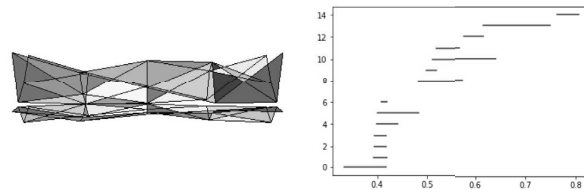
表 1 使用した曲の作曲者と曲数の一覧

変形 Tonnetz の形は作曲者，曲調，曲の長さによって様々なものがある．以下の 2 曲を例として紹介する．またバーコードの計算は [6] によって行った

- 1. ショパン作曲：ワルツ第 17 番変ホ長調



2. ベートーベン作曲：ピアノソナタ第 15 番ニ長調



6 おわりに

本研究では MIDI ファイルから多面体を構成することによって、楽曲の可視化を行った。今後の課題として、この逆問題を考察する。即ち、与えられた多面体からどのような楽曲が生成、構成可能かどうかを考察する。これには多面体を構成するときに、「音の高さの分類」、「リズムの分析」、「コード進行」をよりよく分析することが大事なことになる。その為に Tonnetz を新たにデザインし、楽曲の解析を行う。

参考文献

- [1] 自動作曲システム Orpheus, <http://www.orpheus-music.org/v3/index.php>
- [2] Magenta, <https://magenta.tensorflow.org>
- [3] 高木 将一 中村 啓佑 沼尾 正行, 機械学習の手法を用いた感性の抽出と作曲・編曲への応用, The 15th Annual Conference of Japanese Society for Artificial Intelligence, 2001
- [4] 栢田 幹也, 講座 数学の考え方〈15〉代数的トポロジー, 朝倉書店, 2002
- [5] ピアノ Midi ファイル集, <http://piano.s20.xrea.com/midi/>
- [6] Homcloud, <https://www.wpi-aimr.tohoku.ac.jp/hiraoka-labo/homcloud/>